BIKE ALERT

By

Team #36

David Youmaran (dcy2)

Diego Herrera (dherr4)

Kenneth Kim (kk67)

Final Report for ECE 445, Senior Design, Spring 2025

TA: Aishee Mondal

03 May 2025

Project No. 36

Abstract

This project presents Bike Alert, a real-time theft detection system designed to enhance bicycle security in urban environments. The system uses vibration and Hall-effect sensors to detect tampering or unauthorized access to the lock mechanism. Sensor data is wirelessly transmitted to the user's mobile device via ESP-NOW and Bluetooth Low Energy, providing low latency alerts. An RFID-based secondary lock adds an extra layer of physical security. The companion iOS device displays live sensor data, battery status, and allows users to reset alerts. The final design offers a practical and scalable approach to improving bike security through real-time detection and alert.

Contents

1. Introduction

Bicycle theft remains a persistent and costly problem, especially in urban and campus environments like Champaign-Urbana. On the University of Illinois campus alone, 85 bike thefts were reported between August and mid-November 2023 an increase from 69 during the same period in 2022. Since January 2022, a total of 255 thefts have been documented. While traditional bike locks offer basic physical deterrence, they lack the capability to notify owners in real time, often resulting in undetected thefts until it's too late. This gap in active security underscores the need for an intelligent solution that not only secures the bicycle physically but also actively alerts the user of any suspicious activity.

To address this challenge, we developed Bike Alert, a smart security add-on designed to enhance existing bike locks with real-time monitoring and wireless alert functionality. The system incorporates multiple sensors to detect potential tampering: Hall-effect sensors track the engagement status of the physical lock and the enclosure door, while a custom-designed vibration sensor detects mechanical disturbances. An onboard ESP32 microcontroller gathers this sensor data, interprets tampering events, and transmits alerts using ESP-NOW, a low-power, low-latency wireless protocol. Alerts are relayed to a secondary receiver module, which can then interface with a mobile device for user notification. In addition to detection and alerting, Bike Alert includes an RFID-controlled locking mechanism for secondary physical security, powered by a rechargeable lithium-ion battery for portable, day-long operation.

The following sections detail the design and implementation of the system's key subsystems: power management, tamper detection, locking control, wireless communication, and microcontroller processing. Each section includes design decisions, hardware integration, and verification steps. The report concludes with an evaluation of system performance and functionality, confirming that Bike Alert meets its goal of providing timely theft alerts and reliable locking functionality. Overall, Bike Alert demonstrates a feasible and scalable approach to improving bike security through embedded systems and wireless communication.

2 Design

2.1 High Level Requirements

1. Wireless Communication & Responsiveness

The system shall enable real-time communication between the bike lock and the user's mobile device via ESP-NOW, ensuring secure data transmission within 500 milliseconds over a maximum range of 200 meters between two ESP32 modules.

Reasoning:

ESP-NOW is a low-latency, peer-to-peer protocol ideal for environments where full Wi-Fi infrastructure is not available or would introduce unwanted delay. A transmission latency under 500 ms is crucial to provide timely alerts to the user in case of tampering or unauthorized access. The 200-meter range enables effective outdoor operation, allowing users to monitor and control the lock from a reasonable distance, such as from inside a nearby building or while biking toward their parked bike. This requirement ensures the system is responsive and usable in real-world conditions without relying on a constant internet connection.

2. Power System & Runtime

The system shall operate continuously for at least 24 hours on a single charge using a rechargeable lithium-ion battery under typical usage conditions, including idle monitoring, periodic sensor polling, and occasional communication bursts.

Reasoning:

Since the bike lock is intended for outdoor use, uninterrupted functionality without frequent recharging is critical. A 24-hour runtime guarantees full-day operation even if the user forgets to recharge the device. Power is supplied via a lithium-ion battery, which was chosen for its high energy density and rechargeability. The power delivery is managed by a 12V-to-3.3V buck converter to regulate voltage efficiently while minimizing heat loss compared to linear regulators for the mor demanding Mian board module. This enables reliable powering of the ESP32, sensors, and motor subsystem while maximizing energy efficiency and battery lifespan. A 7.4 to 3.3 LDO for the secondary receiver board only needs to power an ESP32.

3. Tamper Detection & Access Control

The system shall detect tampering via vibration and Hall-effect sensors, trigger an alert within 1 second

of an event, and prevent unauthorized access by controlling the locking mechanism with an RFID module.

Reasoning:

The lock must be secure against unauthorized access and potential theft attempts. A dual-sensor system is employed to ensure redundancy and sensitivity. The vibration sensor detects mechanical disturbances, while the Hall-effect sensors detect changes in magnetic fields that occur when the bike is moved or disassembled. Once tampering is detected, the ESP32 sends a wireless alert via ESP-NOW within 1 second, enabling timely user notification. To complement this, access to the locking mechanism is restricted via an RFID system, which requires a pre-authorized tag to unlock the device. This multifaceted approach enhances the system's security while maintaining ease of use for the owner

2.2 Block Diagram

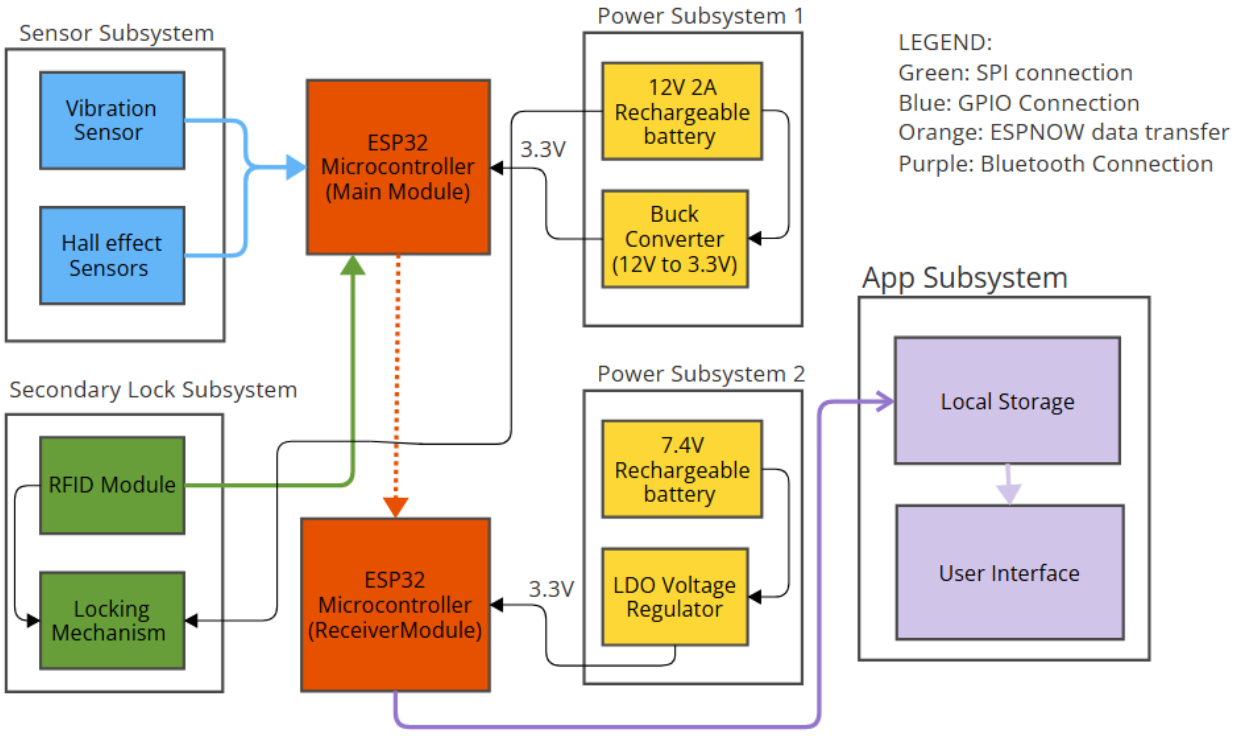

Figure 1: Block Diagram

The block diagram depicts the system design of Bike Alert consisting of 5 main subsystems that follow:

1. Power Subsystems
    - Provides stable regulated voltage for each board
    - 12V 2.6AH battery on main board along with a buck converter circuit to step down to 3.3v for the rest of the electronics
    - 7.4v 2.6Ah battery for the secondary receiver board with an LDO step down circuit to provide the esp32 with 3.3V
2. Sensor Subsystem
    - Detects vibrations and tampering, as well as the engagement status of the physical lock
    - Vibrations are detected when the vibration signal amplitude is greater than the threshold set by a potentiometer. These values are compared as inputs to a comparator and outputs are either high or low
    - 2 Digital Unipolar Hall Sensors are used. A high sensitivity sensor is used for determining lock engagement/disengagement. A low sensitivity sensor is used as part of the rail/tampering system. Both sensors change states at certain magnetic field strengths
3. Locking Subsystem
    - Provides secondary locking mechanism that is controlled via RFID for user convenience
    - Uses a motor with an encoder for precise and consistent movement of locking mechanism.
    - Motor driver to control and drive the motor.
4. Microcontroller subsystem
    - Brain of the main board
    - Processes and manages data received from sensor subsystem
    - Controls logic for locking subsystem tracks locked state and sends motor signals accordingly
5. Application subsystem
    - Provides real-time alerts and haptic feedback directly to your phone during bicycle theft
    - Establishes and manages BLE communication with the receiver module
    - Displays battery monitoring for both ESP32 modules

2.3 Subsystem Design

2.3.1 Power Subsystem

The Power Subsystem is responsible for supplying regulated voltage and current to all active components of the Bike Alert system, including the ESP32, motor, RFID scanner, sensors, and LEDs. The design utilizes a 12V replaceable Li-Ion battery pack as the main power source, stepped down to 3.3V using a 12V to 3.3V buck converter. This buck converter is preferred over an LDO regulator due to its higher efficiency under load, especially important for extending battery life during prolonged active states. The power circuit used for the secondary reviver board was a simple LDO circuit with a Shockey diode given that it only needed to power the ESP32. Initially we had chosen a battery voltage of 3.7V for the reviver but had issues since the voltage difference was too low to convert to 3.3v via the LDO selected, a simple workout to this was to upgrade the battery voltage to 7.4v and ended up working just fine for our purpose without any heating issues or major inefficiencies. Both of these circuits can be found in appendix B figures 3 and 4.

The power circuit also incorporates input capacitors and an output inductor to filter voltage ripple and stabilize power delivery during transient loads (e.g., motor actuation or wireless transmission). All components were selected with compatibility to ESP32 logic levels and current demands in mind. Additionally, a battery monitoring circuit based on a resistive divider and analog sensing via the ESP32 was implemented, allowing real-time voltage tracking to estimate remaining battery life.

The main PCB routes power to all subsystems and uses a screw terminal with barrel connector manual control. Since the ESP32 operates at 3.3V logic and peripherals like the RC522 RFID scanner require similar voltages, the buck converter output ensures compatibility across the board.

To monitor battery health in real time, an analog-to-digital converter (ADC) circuit was integrated into the power subsystem using the ESP32's onboard ADC. A resistive voltage divider scales down the input battery voltage to a level safe for the ESP32's ADC pin (maximum 3.3V). The scaled voltage is then sampled using the ADC, which returns a 12-bit digital value between 0 and 4095 corresponding to the analog voltage between 0 and 3.3V. This value is then mathematically scaled back up to reconstruct the original battery voltage. From this, the system can estimate battery percentage and make decisions such as triggering a low battery warning or preparing for shutdown. This approach is efficient, low-power, and does not require any dedicated battery management IC. The ADC pin used was GPIO8, and the voltage divider used resistors chosen to safely translate the maximum battery voltage (e.g., 12V) into a

measurable voltage under 3.3V. The resulting equation used to reconstruct the input voltage from the ADC value is shown below.

$$\boldsymbol{Vin} = \left(\frac{ADC_{value}}{4095}\right) * V_{ref} * (\frac{R_1 + R_2}{R_2})$$

Where: ADC is the raw 12 bit ADC output, Vref is the ADC reference voltage of 3.3, R1 and R2 or respective voltage divider used to step down the battery voltage, and Vin is the battery voltage before stepping down.

Verification:

To confirm that the power subsystem can sustain the Bike Alert system for at least 24 hours, a combination of real-world testing and quantitative analysis was conducted for both the 7.4V battery powering the receiver PCB and the 12V battery powering the main PCB. During testing, the system was activated under normal operating conditions and voltage measurements were recorded periodically over time. This included continuous operation of the ESP32 microcontrollers, passive and active sensor circuits (vibration, Hall-effect), and intermittent motor and RFID usage. For both batteries, actual voltage data was collected until approximately 65% battery life was left. This ensured that most of the battery discharge behavior was grounded in observation rather than theoretical modeling. The voltage profiles over time demonstrated consistent, stable behavior with during this operating window.

To estimate battery behavior beyond 65% discharge, a nonlinear model was used to emulate the characteristic tapering of lithium-ion batteries toward their end-of-life discharge curves. For the 7.4V battery, a curve exponent of 2.5 was selected to reflect moderate tapering, while the 12V battery was modeled with a slightly steeper exponent (3.0), reflecting its higher-load environment. The resulting discharge curves showed smooth tapering from 100% down to 0% and were consistent with typical behavior based on manufacturer datasheets. This modeling extended the measured dataset to a full 24-

hour window, giving a complete picture of expected system endurance under realistic load.

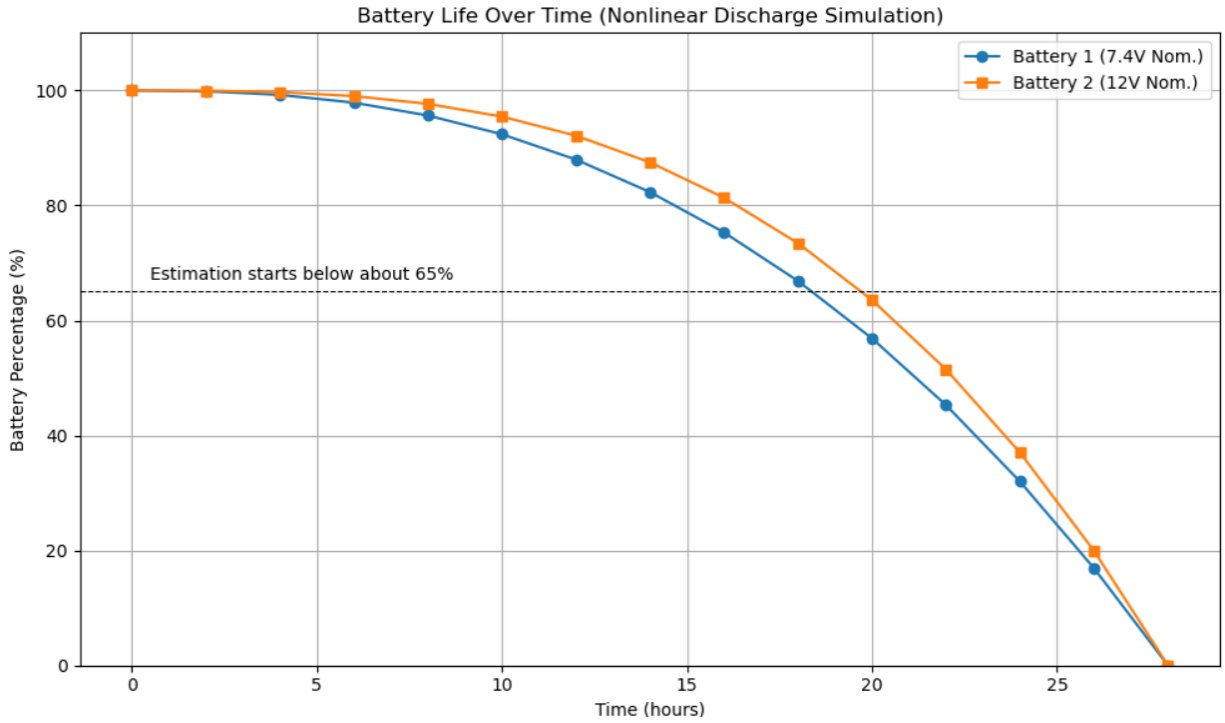

Figure 2 Battery Voltage graph (See appendix B for table)

In this testing, maximum energy consumption estimates were computed for both subsystems to support the discharge of data. For the receiver board, the ESP32 microcontroller was assumed to draw ~180 mA on average at 3.3V due to the constant use of ESP-NOW for data reception. This results in a power draw of 0.594 W, and a daily energy requirement of 14.26 Wh. Accounting for ~85% efficiency of the LDO used to step down from 7.4V, the actual battery-side energy consumption is ~16.78 Wh/day. Given the battery's rated capacity of 2.6Ah at 7.4V (19.24 Wh total), this leaves a buffer of approximately 2.46 Wh, or ~12.8%, allowing for an estimated operational time of ~27.5 hours. This confirms that the receiver system can reliably function for 24 hours with margin.

For the main board, which includes a broader set of peripherals and higher baseline current draw, power consumption was similarly analyzed. The ESP32 is estimated to draw ~240 mA at 3.3V, consuming 0.792 W continuously. The RFID module draws ~26 mA, and the vibration sensor an additional 15 mA, contributing 0.0858 W and 0.0495 W respectively. With these components running 24/7, and the motor drawing 0.15 A at 12V for about 3 seconds per actuation, estimated six times per day, the total daily consumption was calculated to be ~22.78 Wh. The selected 12V, 2.6Ah battery has a total capacity of 31.2 Wh, leaving a buffer of ~8.4 Wh, translating to ~32 hours of continuous operation.

Together, these measures and calculations provide comprehensive evidence that the selected power sources are more than adequate to sustain the Bike Alert system for the target 24-hour operating window.

2.3.2 Sensor Subsystem

The sensor subsystem provides the main ESP32 with sensor data for the 3 sensors used in this project. One vibration sensor and 2 Hall Effect sensors are used. Initial testing of the vibration sensor was done on a SW-420 module [5]. Figure 10 in Appendix B shows the vibration sensor circuit built for our design. It consists of a vibration switch, 10k potentiometer, LM393 comparator, and some resistors and capacitors. A 10k pullup resistor is used between +3.3V and LM393 pin 1 (output). When a vibration occurs, the vibration sensor sends an electrical signal as input to LM393 pin 3; if the amplitude of this signal is greater than the one set by the 10k potentiometer, then the circuit outputs a high state [10]. The potentiometer was tuned to 4.3k. The Hall Effect sensors used in this design are of the digital unipolar variety. A high sensitivity sensor (SS441R) is located near the physical lock and is aligned head on with a neodymium magnet. This sensor serves as the lock engagement/disengagement sensor. While the lock is clamped down by a hook, the lock is not able to move, and the serial monitor shows that the lock is engaged. When the user scans their RFID tag, the hook moves away from the lock and the lock can then move freely. The lock is then able to disengage. Specific details about the RFID and the hook mechanism can be found in sections 2.3.3 and 2.4, respectively. The second Hall Effect sensor used is for tampering detection. A low sensitivity sensor (SS449R) is used for this application to avoid false alerts. A rail system is used in which a neodymium magnet is free to move along a plastic insert on the main PCB enclosure lid. The rail system is described in more detail in section 2.4. Whenever the magnet and sensor are lined up, tampering is detected. Tampering is also detected when the magnet moves past the sensor but at a slower rate since this system is meant to detect harsher break in attempts. As the magnet passes by, the magnetic field strength between the sensor and magnet changes rapidly, so the low sensitivity sensor only picks up on larger changes in the magnetic field, which makes the system reliable for only detecting tampering for harsher break in attempts [4]. Figure 9 in Appendix B shows the distance away from the magnet each sensitivity of Hall Effect Sensor considered triggers.

2.3.3 Locking Subsystem

The locking subsystem provides mechanical actuation and control logic for physically securing and releasing the bike. It consists of a brushed DC motor with an attached quadrature encoder, an RC522

RFID module, and supporting circuitry to drive the motor and interpret feedback signals. The system engages or disengages a hook-style deadbolt based on RFID authentication, all coordinated by an ESP32 microcontroller.

The primary goal of the locking subsystem is to allow the user to unlock the bike using a valid RFID tag. When the tag is scanned, the motor rotates in the appropriate direction to engage or retract the locking mechanism. A brushed DC motor with a built-in quadrature encoder was selected due to its affordability and ease of integration. The encoder outputs two channel signals (A and B), which are fed into interrupt-capable GPIO pins on the ESP32. This setup allows real-time tracking of the motor shaft's movement.

Rather than using a stepper motor, a simpler position tracking system was implemented. The encoder increments or decrements a global encoderPos variable depending on the motor's direction. When locking or unlocking is triggered, the motor rotates until the encoderPos reaches a predetermined threshold (lockDistance), which corresponds to the full engagement or retraction of the hook mechanism. The only problem encountered with this system was the quality of the chosen motor, we had an issue where the gears slipped, preventing the motor from working properly. The solution we implemented was to open the gearbox of the motor and fix the gear back into place along with adding bearing grease to the gears. This solution proved to be quite effective as we didn’t have any motor issues after that point and the motor worked better than it came from the manufacturer.

The motor is driven by a dual H-bridge driver (L298N), allowing for full forward/reverse control and dynamic braking. The RFID module communicates with the ESP32 using SPI and signals a valid unlock condition when a registered tag is scanned. The locking sequence is only initiated if the tag is authenticated successfully.

The subsystem was implemented on the main PCB, which hosts the ESP32, motor driver circuitry, and logic-level conversion needed for the RFID reader. Care was taken to assign interrupt-capable GPIO pins to the encoder channels to ensure reliable pulse counting. The system logic is programmed to rotate the motor for a set distance based on encoder counts. No limit switches were used, so the encoder feedback is critical to prevent the motor from stalling or running indefinitely.

The lockDistance constant was introduced to define how far the motor should turn (in terms of encoder counts) during a full lock or unlock operation. This value is specific to the mechanical linkage between the motor and the locking hook including gear ratios, backlash, and physical travel limits. For reliability,

the system also includes a short delay after motion completion and forcibly stops the motor by shorting both terminals via the H-bridge to prevent drift or overshoot.

Verification

Since the encoder position (encoderPos) does not directly correspond to real-world units like degrees or centimeters and because motor movement depends on torque, load, and geartrain imperfections a trial-and-error method was used to calibrate the lockDistance.

To calibrate the motor travel:

1. A Serial.println(encoderPos) statement was added to the firmware.
2. The system was powered, and the RFID scan sequence was triggered repeatedly while observing how the encoderPos value changed.
3. The physical hook was visually monitored during each locking and unlocking attempt.
4. The value of lockDistance was iteratively adjusted until the hook fully engaged or retracted without over-driving or binding the mechanism.

The final calibrated value of lockDistance = 3050 was determined experimentally. This means that each lock or unlock cycle rotates the motor until the encoderPos increases or decreases by 3050 counts from its initial state. This value is unique to this specific prototype and may differ for future iterations due to slight changes in mechanical tolerances or motor mounting. This approach worked reliably and eliminated the need for hardware limit switches or more complex control algorithms. Although it doesn't map to a physical measurement like degrees or millimeters, it provides consistent and safe operation within the system's mechanical limits. Images of the print serial motor showing the encoder position can be found in appendix B

The locking subsystem was tested over multiple cycles, and in all cases, the hook reached its travel limit cleanly without mechanical interference. The encoder feedback ensured repeatable motion control, and the system successfully responded to all valid RFID scans with the expected motor behavior. Overall, the trial-based calibration method provided a robust and flexible way to implement locking without excessive hardware overhead.

2.3.4 Microcontroller Subsystem

The microcontroller subsystem serves as the central processing unit for the entire main board, orchestrating all sensing, decision-making, and communication tasks within the Bike Alert system. At its core is an ESP32 microcontroller, chosen for its robust performance, native support for ESP-NOW a low-latency, peer-to-peer communication protocol ideal for real-time wireless updates between the lock module and the receiver unit.

This subsystem performs several critical functions. It continuously collects data from multiple sensors integrated into the lock. The vibration sensor and two Hall-effect sensors serve as the system's primary tamper detection tools. These sensors are connected to digital GPIO inputs on the ESP32 and are polled regularly or handled via interrupts to minimize delay in detecting tampering. In addition to security sensing, the ESP32 also monitors the system's power state. A battery voltage monitoring circuit comprising a resistive voltage divider connected to one of the ESP32's analog input pins allows the microcontroller to sample and report real-time battery levels. This data ensures the user can receive proactive alerts before power levels fall too low.

Beyond sensing, the microcontroller is also responsible for managing the locking mechanism. It interfaces with an RC522 RFID scanner via SPI to verify RFID tag input. Upon detecting a valid tag, the ESP32 executes locking or unlocking logic by sending direction and enable signals to the motor driver, which then actuates the encoder motor responsible for turning the physical hook lock. To control the motor accurately, the ESP32 monitors encoder feedback using interrupts on two dedicated GPIOs. Since the encoder's position output does not directly correlate with angle or RPM, the system uses a trial-and-error calibrated constant (lockDistance) to determine the necessary count value for full engagement or retraction of the hook. This ensures the locking motor stops at precisely the right moment without overdriving or stalling.

Once sensor data is processed and actions are taken, ESP32 sends relevant information to the receiver board using ESP-NOW. The transmitted packet includes the current battery voltage, tamper alerts, and lock status flags. This wireless communication is designed to operate with a latency under 500 milliseconds and a range of up to 200 meters, keeping the user reliably informed in real time.

Verification:

The microcontroller subsystem was verified in two main stages. First, a continuity test was performed across all relevant pins and connections to ensure proper electrical integration between the ESP32 and

connected components such as the motor driver, sensors, RFID scanner, and voltage monitoring circuit. After confirming hardware integrity, functional verification was performed using known-working code from earlier testing. This included uploading firmware modules for ESP-NOW transmission, RFID reading, and encoder-based motor control. Serial output was used to monitor system behavior in real time, allowing confirmation that sensor inputs, motor actuation, and data transmission were functioning as intended. These steps demonstrated that the ESP32 successfully managed all key subsystems and could be reliably programmed and operated on the custom PCB.

2.3.5 Application Subsystem

The Application Subsystem is responsible for interfacing with the user and presenting real-time alerts and status updates from the Bike Alert system. This subsystem centers around a native iOS application developed in Swift, leveraging the CoreBluetooth framework to communicate with the receiver module via Bluetooth Low Energy (BLE).

The architecture of the system is built around a dual microcontroller model, where a primary ESP32 microcontroller located on the bike transmits data wirelessly to a second receiver ESP32 using ESP-NOW, a lightweight low latency protocol developed by Espressif that provides the ability to transmit data over distances up to 200 meters without requiring WiFi, making it ideal for a bike lock. Sensor data generated by the main ESP32 is encapsulated into a struct and transmitted as bytes on an ESP-NOW payload. These packets are received by the BLE server ESP32, which updates its general attribute (GATT) characteristic values accordingly. The BLE server then advertises itself to the mobile, which acts as a BLE central, scanning for and connecting to peripherals broadcasting the predefined UUIDs. These UUIDs must match up between the mobile application and the BLE server, or else it will not connect.

Upon connection, the app discovers the BLE service and characteristic, then subscribes to notifications to receive live updates. BLE characteristic values are parsed and displayed in a structured SwiftUI interface comprising three sections: connection status, sensor alerts and battery status. If multiple vibration/tampering events are triggered in a 0.5 second window of time, it remains active for 2-3 seconds to prevent any missed events caused by brief fluctuations. Users can also manually reset these alerts after confirming the bike is secure, via a dedicated button in the UI.

Battery voltage data from both ESP32s is computed using onboard analog-to-digital converters and voltage divider circuits, then sent over ESP-NOW and BLE. The mobile app applies an exponential moving average filter to smooth voltage readings and suppress noise from analog drift and sampling jitter.

Battery levels are displayed as progress bars with numerical percentage labels, and notifications are sent when the battery drops below 10%.

Verification: To ensure real-time responsiveness and validate signal strength over large distances, a series of range tests were conducted using two ESP32 developer boards. One ESP32 would transmit data packets using fake data via ESP-NOW to a receiver ESP32 module. Once the data was received, the second ESP32 measured and logged the Received Signal Strength Indicator (RSSI) values associated with each received packet. Data was collected outdoors over 4 distances: 5 meters, 50 meters, 100 meters, and 200 meters. The RSSI logs were parsed, grouped by distance, and visualized using box plots to assess signal stability and loss characteristics over range. The box plot can be found under Appendix B Figure 8 and the results are summarized below:

- At 5 meters, the RSSI ranged between -23 dBm and -27dBm with minimal variance, indicating excellent signal strength quality
- At 50 meters, values ranged between -52 dBm and -71 dBm, still well within reliable reception bounds with <1% packet loss observed.
- At 100 meters, the RSSI averaged around -75 dBm, with slightly wider distributions, indicating some minor packet loss and drop in strength
- At 200 meters, the RSSI dropped closer to -87 dBm with increased spread, starting to show greater signs of packet loss, which is expected.

These range tests confirmed that the ESP-NOW communication remains stable up to around 180-200 meters, aligning with Espressif's claims. The range can further be improved by adding an antenna to the module, or by putting the ESP32s in long range mode, which we chose against due to the range being sufficient in the regular mode, as well as a higher power consumption.

2.4 Physical Implementation

The physical implementation of the Bike Alert system began with the design of the secondary locking mechanism. Initial concept sketches were created to facilitate discussions with the machine shop, allowing us to explore fabrication options and constraints. Through collaborative iteration, we finalized a locking mechanism consisting of a custom-fabricated metal hook mounted onto the shaft of a DC motor. The hook was attached via a machined circular adapter, enabling unidirectional rotation of the hook to engage and disengage the lock. This mechanical motion formed the basis of the system's motor-driven locking subsystem.

In parallel with the locking mechanism, efforts were made to design and fabricate the mounting solution for the electronics enclosure. This included careful measurement of internal components such as the PCB and battery to ensure proper spatial arrangement and wire routing within the selected housing. The enclosure was then mechanically mounted onto the locking assembly using machined supports, resulting in a compact and robust unit. The final integrated assembly is shown in Figure X.

The next stage involved mounting the system's sensors and corresponding magnets to enable proper tamper detection and lock status monitoring. The tamper detection system was implemented using a custom plastic insert fabricated by the machine shop, which fit into the lid of the enclosure and housed a freely moving magnet. This magnet would shift in response to vibration, triggering the Hall-effect sensor placed on the PCB to detect tampering events. This subsystem is illustrated in Figure X.

Additional components, including external magnets and the Hall-effect sensors, were mounted using adhesives such as super glue and secured with tape where necessary. The sensors were positioned precisely to ensure consistent operation regardless of the direction in which the lock was closed. Final sensor placement was refined through a combination of measurement, prototyping, and trial-and-error adjustments. Cable management was finalized using heat shrink tubing to maintain electrical safety and organization within the enclosure.

Overall, collaboration with the machine shop was essential in fabricating the custom mechanical components and ensuring a professional, functional integration of electronics and mechanical systems. All sensors performed as intended during final testing, with accurate alert triggering and reliable locking operation.

3. Cost And Schedule

3.1 Parts

The Bill of Materials (BOM) as well as the associated cost and labor can be found in Appendix A.

3.2 Schedule

The schedule followed in the Bike Alert project can be found in Appendix A.

4. Conclusion

4.1 Accomplishments

The Bike Alert project achieved its primary objectives, culminating in a fully functional prototype that successfully integrates all subsystems sensors, wireless communication, and a secondary locking mechanism into a cohesive and responsive security solution. Our greatest accomplishment was the seamless coordination between these components to provide real-time alerts to a bike owner in the event of tampering or theft. This outcome reflects the effectiveness of our system architecture and our ability to implement both hardware and software systems in tandem.

The final product met all the functional and performance requirements outlined at the start of the semester, including accurate sensor detection, reliable ESP-NOW communication, and a user-friendly RFID-controlled locking mechanism. The secondary locking system operated as intended, reliably engaging and disengaging with motor control and enhancing the overall security of the system. The project demonstrated successful integration, solid electrical design, and practical feasibility for real-world application.

4.2 Uncertainties and Future Work

Although the final prototype performed as expected, several areas for refinement remain primarily in the domain of mechanical design and physical robustness. The most notable area for future improvement involves the structural implementation of the locking mechanism. While the hook-based locking design functioned effectively and could not be disengaged by force during testing, the connection between the motor and the rotating shaft could be improved. Specifically, the motor shaft attachment could benefit from a more rigid, machined coupling solution, or potentially custom housing to better support axial loads and reduce mechanical stress.

Additionally, the enclosure housing the motor and locking mechanism left some components exposed, which could pose a vulnerability in a real-world theft scenario. Future iterations of the design should consider a more durable and secure casing that fully encapsulates the mechanical assembly, further improving the physical integrity and security of the system.

Despite these mechanical limitations, the electronic components performed reliably, and any physical breach would still trigger a tamper alert via the vibration and Hall-effect sensors. Nonetheless, addressing these mechanical shortcomings would elevate the system's security and durability for everyday use.

Looking ahead, future development could include enhancements such as GPS integration for real-time location tracking that could be integrated into our working application. These improvements, along with strengthened mechanical design, would significantly increase the utility, robustness, and commercial viability of the Bike Alert system.

4.3 Ethical considerations

The Bike Alert system was developed with careful consideration of ethical responsibilities and safety standards to ensure its deployment is secure, responsible, and in the best interest of users and the broader public. The project adheres to the IEEE Code of Ethics by prioritizing user safety, security, and privacy while actively working to reduce theft and promote responsible engineering practices [2].

A central ethical concern in modern security systems is privacy protection. Bike Alert has been designed to avoid the collection, storage, or transmission of any personally identifiable information. The system solely transmits tamper alerts from the main locking unit to the user’s receiver module via ESP-NOW, ensuring that the user remains informed without compromising personal data or enabling tracking. This approach upholds ethical standards in data handling and protects users from potential surveillance or data misuse.

Safety in operation was another primary design goal. The motor components used to control the physical locking mechanism are chosen and configured to minimize any risk of accidental injury during use. Mechanical movements are restricted to slow and deliberate actions. On the electrical side the PCB was designed to prevent potential short circuits. Heat dissipation and component tolerances were carefully evaluated to avoid fires or failures during typical use.

The system also considers relevant regulatory compliance. As the device uses ESP-NOW, which operates over the 2.4 GHz ISM band, it must comply with FCC Part 15 rules on unlicensed radio transmitters. While formal certification is outside the scope of this academic project, all components and transmission characteristics were selected to fall within accepted regulatory norms. In addition, the rechargeable lithium-ion battery was chosen with regard to UN safety guidelines [3], which govern battery testing and shipping standards. The battery's voltage range, charging circuitry, and protection features were all designed to prevent overcharging, overheating, or leakage.

To ensure ethical use, the system includes misuse prevention mechanisms that allow only authorized users to unlock or disable the device. This is achieved through an RFID-based authentication system that checks for valid tags before activating the lock. Without a registered RFID tag, the system remains locked and continues to monitor for tampering, thereby preventing unauthorized access and reducing the risk of false disarming by third parties.

By proactively addressing these ethical and safety concerns, the Bike Alert system demonstrates a strong commitment to user protection, privacy, and regulatory responsibility. The result is a solution that not only deters theft but also upholds the standards of ethical engineering and safe product design.

References

[1] CU-CitizenAccess, “Bike thefts surge on University of Illinois campus, but online reporting system aids police investigations,” Jan. 2024. [Online]. Available: https://cu-citizenaccess.org/2024/01/bike-thefts-surge-on-university-of-illinois-campus-but-online-reporting-system-aids-police-investigations/. [Accessed: Aug. 30, 2025].

[2] IEEE, “IEEE Code of Ethics.” [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: Aug. 10, 2025].

[3] United Nations, “UN Manual of Tests and Criteria, Section 38.3.” [Online]. Available: https://unece.org/transport/standards/transport/dangerous-goods/un-manual-tests-and-criteria-rev8-2023 [Accessed: Aug. 30, 2025]

[4] Honeywell, Unipolar Digital Hall-Effect Sensor ICs, 2014. [Online]. Available: https://prod-edam.honeywell.com/content/dam/honeywell-edam/sps/siot/de-de/products/sensors/magnetic-sensors/unipolar-position-sensor-ics/common/documents/sps-siot-ss340rt-ss440r-unipolar-digital-hall-effect-sensor-ics-productsheet-005909-4-en-ciid-45262.pdf?download=false. [Accessed: Mar. 28, 2025].

[5] Components101, "SW-420 Vibration Sensor Module," *Components101*, 2022. [Online]. Available: https://components101.com/sensors/sw-420-vibration-sensor-module. [Accessed: Aug. 10, 2025]

[6] RFID Technology, *RFID Module 4411 Datasheet*, 2021. [Online]. Available: https://mm.digikey.com/.../4411_CN0090%20other%20related%20document%20%281%29.pdf. [Accessed: Aug. 10, 2025]

[7] Tronsun Motor, *DC Brush Motor with Encoder Datasheet*, 2019. [Online]. Available: https://www.tronsunmotor.com/.../e78fcf93ed604a64c69852b5db49a03f.pdf. [Accessed: Aug. 28, 2025]

[8] Espressif Systems, *ESP32-S3-WROOM-1/WROOM-1U Datasheet*, 2021. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32-s3-wroom-1_wroom-1u_datasheet_en.pdf. [Accessed: Aug. 28, 2025]

[9] Espressif Systems, *ESP32-S3-DevKitC-1 User Guide*, 2024. [Online]. Available: https://docs.espressif.com/projects/esp-dev-kits/en/latest/esp32s3/esp32-s3-devkitc-1/user_guide.html#hardware-reference. [Accessed: May 1, 2025].

[10] SunFounder, *Vibration Sensor Module (SW-420)*, 2025. [Online]. Available: https://docs.sunfounder.com/projects/ultimate-sensor-kit/en/latest/components_basic/04-component_vibration.html. [Accessed: Aug.29 2025].

[11] Random Nerd Tutorials, "Getting Started with ESP-NOW (ESP-32 with Arduino IDE)," [Online]. Available: https://randomnerdtutorials.com/esp-now-esp32-arduino-ide/. [Accessed: Apr. 2, 2025].

[12] N. Kolban, "ESP32 BLE Arduino," *GitHub Repository*. [Online]. Available: https://github.com/nkolban/ESP32_BLE_Arduino. [Accessed: Apr. 2, 2025].

[13] GUID Generator, "Online GUID Generator Tool." [Online]. Available: https://www.guidgenerator.com/. [Accessed: Apr. 1, 2025].

[14] R. Santos and S. A. Silva, "ESP32 BLE Server and Client (Bluetooth Low Energy)," *Random Nerd Tutorials*. [Online]. Available: https://randomnerdtutorials.com/esp32-ble-server-client/. [Accessed: Apr. 2, 2025].

[15] Apple Inc., "SwiftUI," *Apple Developer Documentation*. [Online]. Available: https://developer.apple.com/documentation/swiftui. [Accessed: Apr. 1, 2025].

[16] Apple Inc., "CoreBluetooth," *Apple Developer Documentation*. [Online]. Available: https://developer.apple.com/documentation/corebluetooth. [Accessed: Apr. 1, 2025]

Appendix A Requirement and Verification and Bill of Materials

Description	Manufacturer	QTY	Extended Price	Link
DC Gear Motor 12V Low Speed 10RPM Encoder Metal Gearmotor	Bemonoc	1	$15.88	Link
10PC Male Header Pins,40 pin Header Strip	MCIGICM	1	$4.99	Link
12V Rechargeable Lithium-ion Battery Pack with Charger	Rapthor	1	$23.89	Link
2PC RFID Reader Writer RC522 Sensor Module	WWZMDiB	1	$6.99	Link
5PC SW-420 Vibration Sensor Module	EC Buying	1	$4.99	Link
CONN PWR JACK 2X5.5MM SOLDER	Same Sky	1	$0.76	Link
CONN HEADER VERT 2POS 2.5MM	JST Sales America Inc.	1	$0.10	Link
SLOW VIBRATION SENSOR SWITCH (HA	Adafruit Industries LLC	1	$0.95	Link
IC REG BUCK 3.3V 2A TSOT23-6	Diodes Incorporated	1	$1.38	Link
CAP TANT 10UF 20% 10V 1206	KEMET	1	$0.35	Link

SWITCH TACTILE SPST-NO 0.05A 16V	C&K	2	$0.98	Link
KBT 3.7V 3000mAh Rechargeable Li-ion Battery with JST 2.54 2Pin Plug, Charging Cable for Voice Power Amplifier, Speaker	KBT	1	$12.99	Link
3.7V USB Charging Cable XH 2.54mm 2Pin Plug to USB Connector Lithium Battery Charger for Aircraft Helicopter Toys 3.7v Battery Charger Cord	GMBYLBY	1	$4.16	Link
Inductor 4.7UH 4.5A 26 MOHM SMD	Bourns Inc.	1	$0.44	Link
Grand Total - Parts and Labor			$20,891.35	

Week	**Tasks**	**Person**
Week of March 3rd	1. Design Documents are due Thursday. 2. Finish ordering parts used for testing 2. Continue working on circuitry for breadboard demo	1,2,3 - Everyone
Week of March 10th	1. Breadboard Demo Wednesday @ 2 pm	1 - Everyone
Week of March 17th	SPRING BREAK	Nobody
Week of March 24th	1. Unit test Test/Revise PCB 2. Finish MVP for app, get it ready for testing with	1 - Diego, David

	microcontroller	2 - Kenny
Week of March 31st	1. Unit test Test/Revise PCB 2. Test connection between the two ESP-32s	1 - Diego, David 2 - Kenny
Week of April 7th	1. Unit test Test/Revise PCB 2. Finalize testing and add visuals for app	1 - Diego, David 2 - Kenny
Week of April 14th	1. Get Final PCB working 2. Finalize app, do some range tests	1 - Diego, David 2 - Kenny
Week of April 21st	Practice our mock demos, finalize presentations	Everyone
Week of April 28th	Final demo with TA and continue working on final papers and documentation	Everyone
Week of May 5th	Final Presentations	Everyone

Requirement	Verification	Verification status (Y or N)
1. Sensor Subsystem Requirements a. The vibration sensor must distinguish between normal environmental	1. Sensor Subsystem Verifications a. 1a. Use varying degrees of force to shake the sensor and determine what	Y

vibrations and tampering. b. The ESP32 must process sensor data and transmit an alert via ESP-NOW within seconds of detection. c. Hall-effect sensors must detect changes in the magnetic field within desired distance.	amount of force causes the sensor to trigger. 2a. Adjust potentiometer value on vibration sensor circuit by turning knob to determine comparator threshold voltage needed to distinguish between normal vibrations and tampering. b. 1b. Use Arduino IDE to program the Main ESP32 to read sensor data. 2b.Transmit sensor data to Receiving ESP32 via ESP-NOW wireless communication protocol 3b. Measure processing delay by logging sensor input timestamps and comparing them to the timestamps when an alert is transmitted via ESP-NOW. c. 1c. Use a measuring tape to measure the distance the lock disengagement sensor moves away from the magnet mounted on the lock. 2c. Repeat step 1 for tampering detection Hall-effect sensor; in this case, we will use the measuring	

	tape to determine that the sensor detects changes to the magnetic field at desired distance the magnet moves on the rail.	
2. Power Subsystem Requirements a. The 12V to 3.3V buck converter must output a stable 3.3V ± 0.1V at up to 500mA. b. The 3.7V to 3.3V LDO must supply a steady 3.3V output to the receiver ESP32. c. The 12V battery must provide at least 500mA to power the motorized locking mechanism. d. The receiver PCB must function for at least 24 hours on a single battery charge.	2. Power Subsystem Verifications a. Use a multimeter to verify the voltage remains within range under different load conditions. b. Measure output voltage with a multimeter under normal operating conditions. c. Measure current draw while actuating the motor to confirm sufficient supply. d. Perform continuous operation tests to see consumption over time.	Y
3. Locking Subsystem Requirements a. The RFID module must authenticate a valid tag within 500 ms. b. The motor must rotate a precise amount to engage and disengage the lock reliably. c. The motor driver must operate within a 12V ± 5%	3. Locking Subsystem Verifications a. 1a. Test response time by scanning multiple tags and measuring delay with a high-resolution timer. 2a. Conduct security testing with unauthorized tags to ensure proper rejection. 3a. Log successful and failed authentication attempts to	Y

range. d. The ESP32 must properly interpret encoder signals to prevent over-rotation.	verify system reliability. b. 1b. Monitor encoder feedback and compare expected vs. actual motor positions. 2b. Use an oscilloscope to verify PWM signal integrity and stability. 3b. Conduct repeated locking/unlocking cycles and measure any deviation in motor position using high-speed video analysis or an encoder data logger c. 1c. Measure voltage across the motor driver input under varying load conditions to confirm it remains within 11.4V to 12.6V. 2c. Use a current probe to monitor power draw and verify it remains within expected parameters. 3.c Perform mechanical stress tests by simulating lock resistance and ensuring the motor consistently completes its rotation. d. Monitor encoder feedback in real-time and confirm expected pulse counts for	

	each locking/unlocking cycle.	
4. Microcontroller Subsystem Requirements a. Must process sensor inputs within 1 second and transmit alerts via ESP-NOW within 2 seconds of tampering detection. Must operate at 3.3V with a current draw of up to 120mA in active mode and an additional 160mA during wireless transmission b. Must accurately process data from the Hall-effect sensors, vibration sensor, and RFID module and transmit alerts within 2 seconds via ESP-NOW. c. Must generate PWM signals to precisely control the motorized lock and prevent over-rotation. d. Must support Bluetooth Low Energy (BLE) communication with a mobile app, ensuring real-time data exchange.	4. Microcontroller Subsystem Verifications a. 1a. Measure processing delay by logging sensor input timestamps and comparing them to the timestamps when an alert is transmitted. 2.a Use a multimeter to measure voltage stability on the 3.3V rail under various loads. 3.a Monitor current draw using an oscilloscope or multimeter in different operating states (idle, processing, transmitting). b. 1b. Use Arduino IDE to program the Main ESP32 to read sensor data. 2b.Transmit sensor data to Receiving ESP32 via ESP-NOW wireless communication protocol 3b. Measure processing delay by logging sensor input timestamps and comparing them to the	Y

e. Must support communication between the server ESP-32 and the receiver ESP-32.	timestamps when an alert is transmitted via ESP-NOW c. 1c. Monitor encoder feedback and compare expected vs. actual motor positions. 2c. Use an oscilloscope to verify PWM signal integrity and stability. 3c. Conduct repeated locking/unlocking cycles and measure any deviation in motor position d. 1d. Monitor data exchange between the ESP-32 and mobile app, ensure that the ESP is sending data using serial monitoring to print confirmation messages. 2d. Ensure the app has received said data, by checking local storage in real time. 3d. Send arbitrary data to the ESP-32 and check functions return type, should be True if data was sent successfully. 4d. Ensure ESP-32 receives data from the app by using serial monitoring and checking if data printed	

	matches the data sent. e. 1e. Broadcast data from the server ESP-32 to the receiver's MAC address and print out sent confirmations 2e. Print the data from the receiver and ensure it aligns with the data sent from the server	
5. Application Subsystem Requirements a. One ESP-32 must act as a sender, using ESP-NOW Peers to send data to the receiver b. Receiver ESP-32 must send data to the mobile application via Bluetooth c. Mobile App must be able to communicate with the ESP-32 and translate real time data into notification	5. Application Subsystem Verifications a. Use a serial monitor, where the sender ESP-32 prints out confirmation of sending messages, while the receiver prints out the received data/message b. Debug using Apple's CoreBluetooth framework, initialize a CBCentralManager and develop unit tests to ensure that each different packet is sent and received to the mobile application. c. Similar to the above, we can use the CoreBluetooth framework to write unit tests to send data to the ESP-32 and monitor the packets sent to the	Y

	microcontroller and see if they match with the ones we sent.	

Appendix B Supplemental Images

Time (hrs)	Battery 1 Voltage (V)	Battery 1 (%)	Battery 2 Voltage (V)	Battery 2 (%)
0	8.4	100	12.32	100
2	8.39754556	99.86364222	12.31970117	99.96355685
4	8.386115591	99.22864393	12.31760933	99.70845481
6	8.361739067	97.87439263	12.31193149	99.01603499
8	8.321457921	95.63655115	12.30087464	97.66763848
10	8.262792634	92.37736856	12.28264577	95.44460641
12	8.18356348	87.97574891	12.2554519	92.12827988
14	8.081801948	82.32233047	12.2175	87.5
16	7.955698905	75.31660584	12.16699708	81.34110787
18	7.803571086	66.86506031	12.10215015	73.43294461
20	7.623837929	56.87988496	12.02116618	63.55685131
22	7.415004767	45.2780426	11.92225219	51.4941691
24	7.175650154	31.9805641	11.80361516	37.02623907
26	6.904416047	16.91200262	11.6634621	19.93440233
28	6.6	0	11.5	0

Table 1: Table form of Figure 2

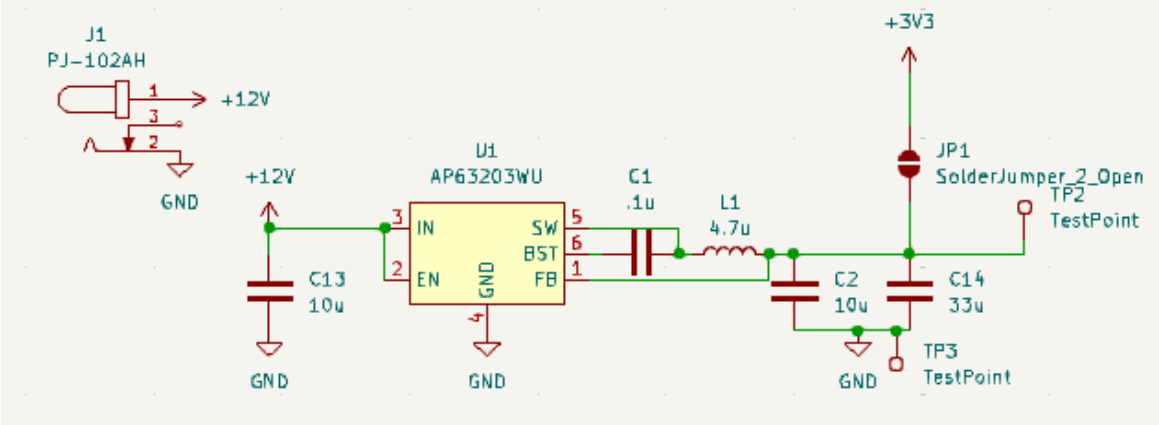

Figure 2: Buck Converter Circuit For Main PCB

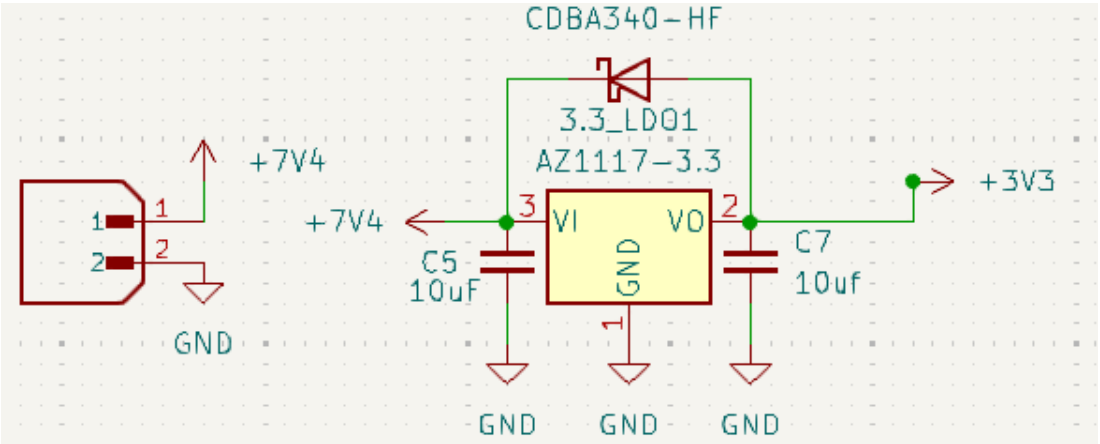

Figure 3: LDO Circuit for Receiver PCB

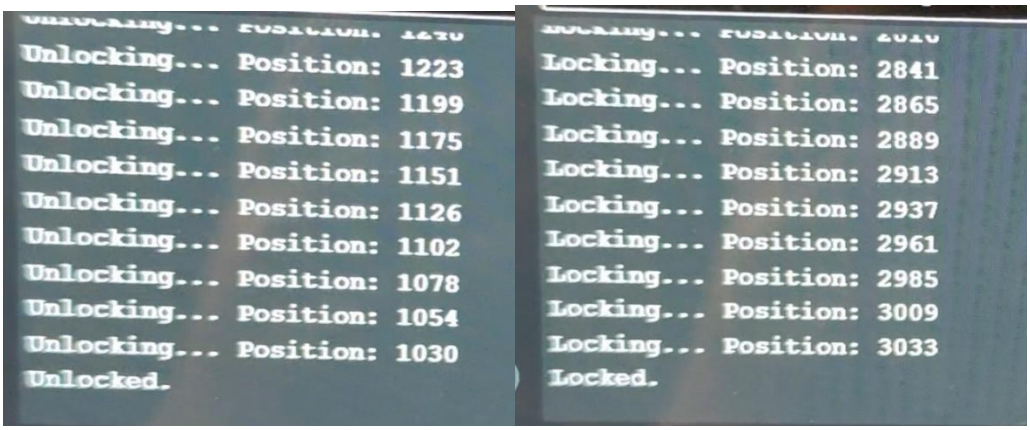

Figure 5: Serial Monitor of Locking Mechanism Verification Prosses

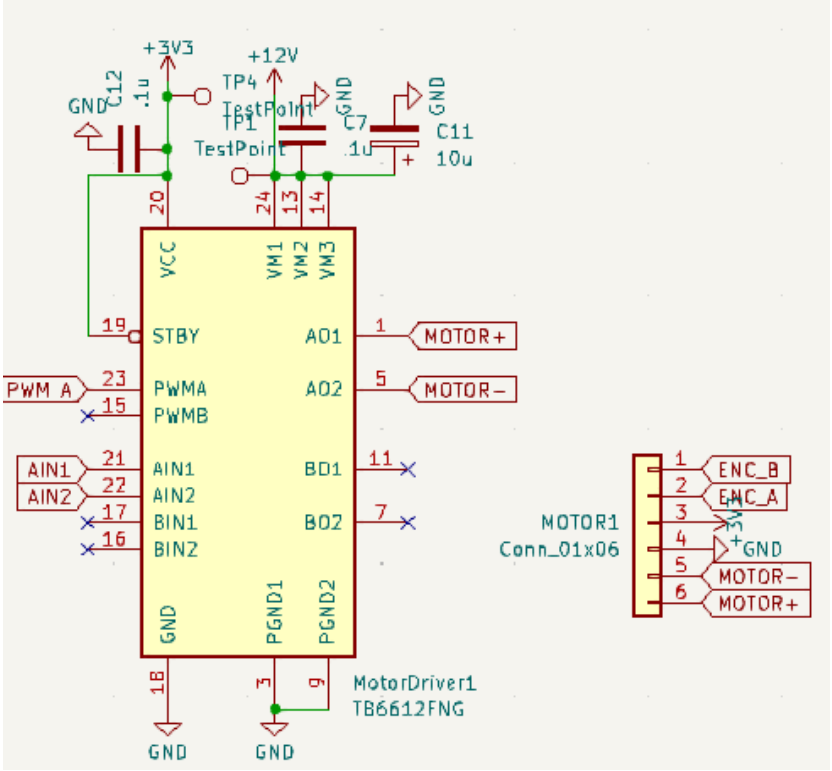

Figure 6: Locking Subsystem Schematic

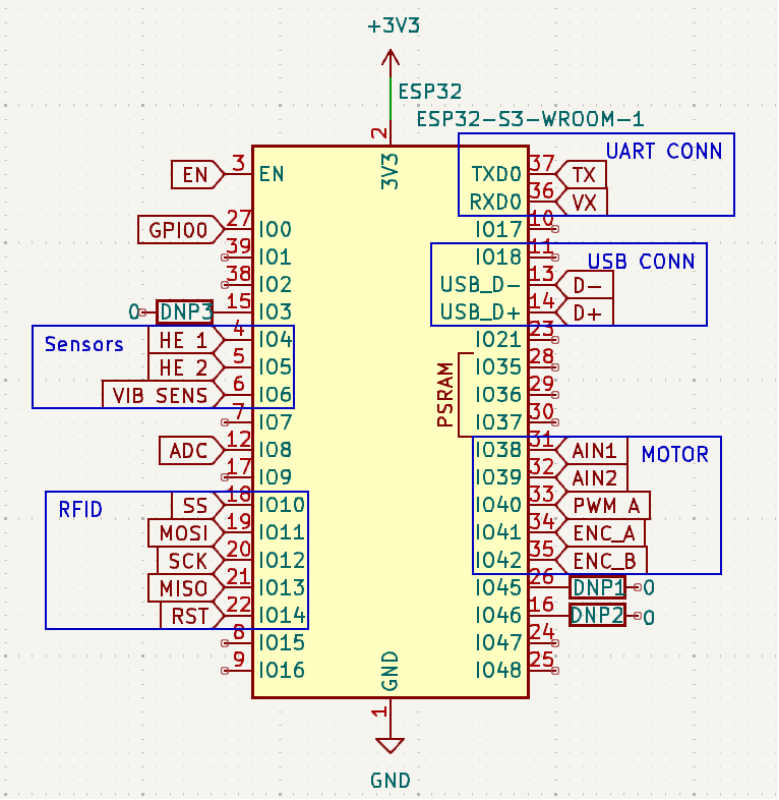

Figure 7: Microcontroller Connections

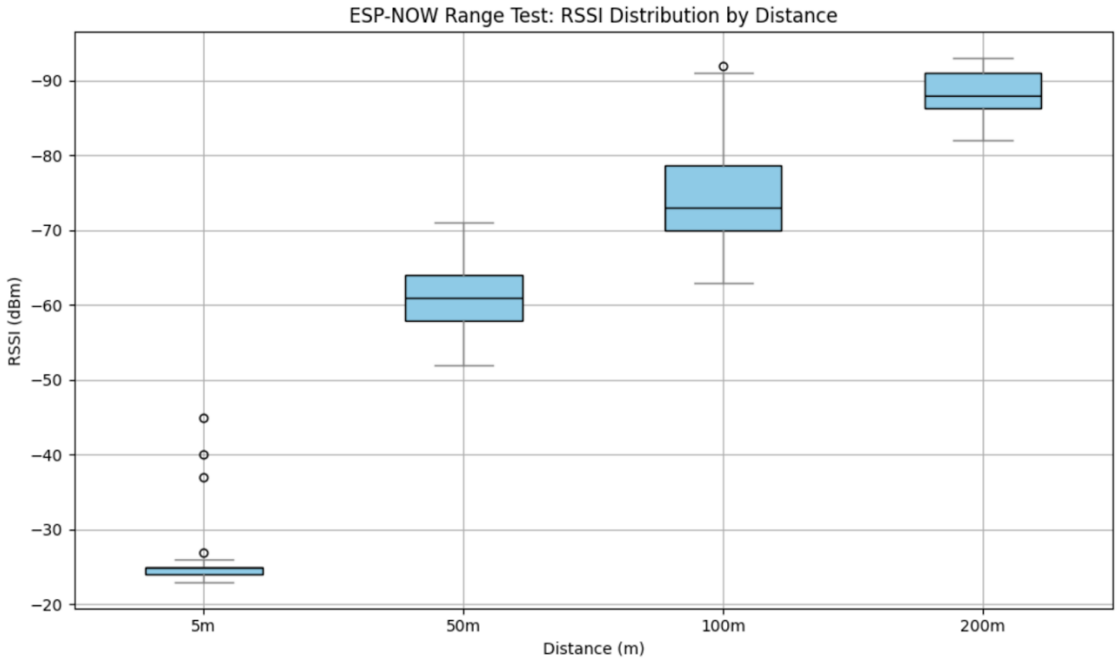

Figure 8: ESP-NOW Range Test

Sensor Model	Distance away from magnet required to trigger (inches)
SS441R (High Sensitivity)	0.625
SS443R (Medium Sensitivity)	0.5
SS449R (Low Sensitivity)	0.25

Figure 9: Hall Effect Sensor Data

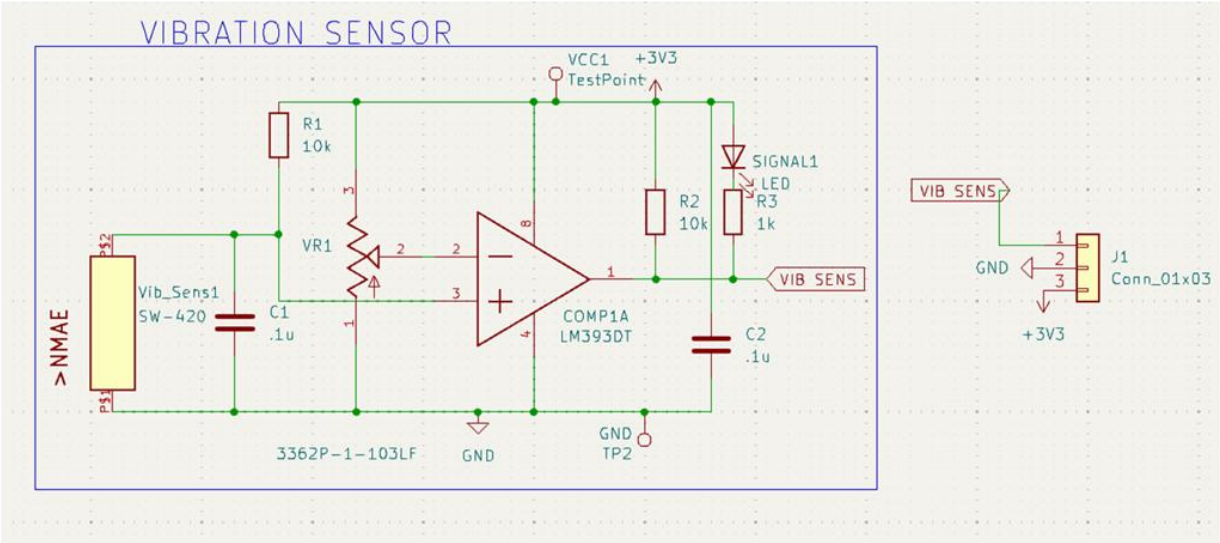

Figure 10: Vibration Sensor Circuit